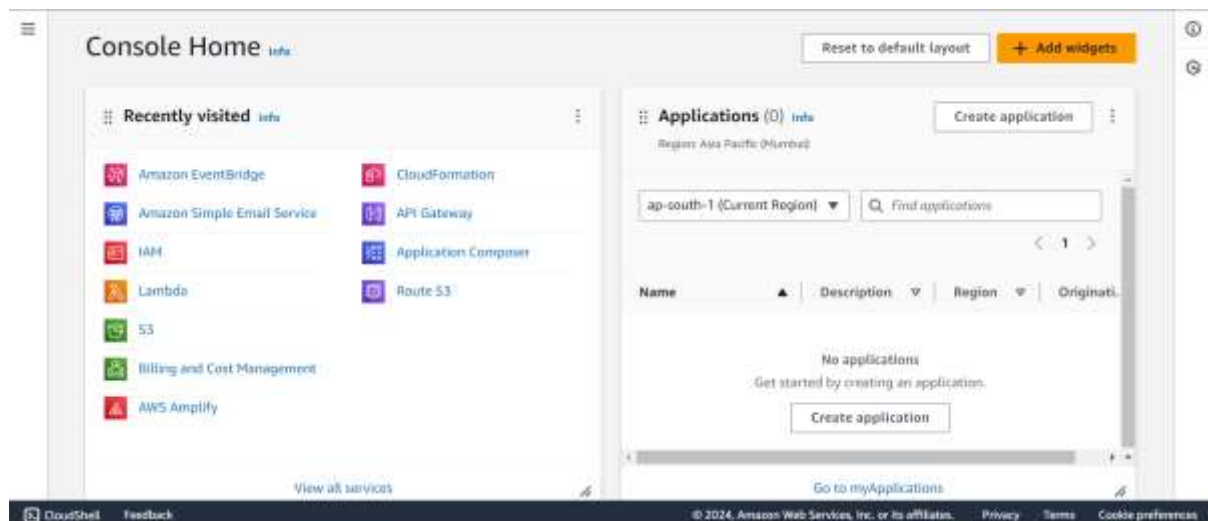


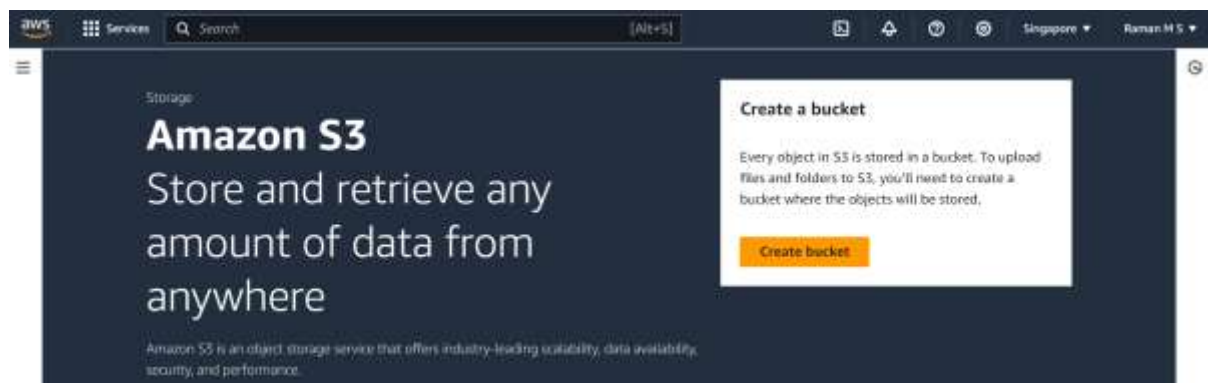
Build a Serverless Email Marketing Application Using AWS

The Services used in this projects
(S3,SES,AWS LambdaFunction,AWS Event Bridge)



Step 1: Create an S3 Bucket

- Navigate to S3 in the AWS Console.
- Click the “Create bucket” button.



- Add a globally unique name for your S3 bucket.

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
Asia Pacific (Mumbai) ap-south-1

Bucket name [Info](#)
tft-email-marketing2

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied

[Choose bucket](#)

Format: s3://bucket/prefix

Object Ownership [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

- Untick “**Block all public access**” -(it enables public access to your bucket’s contents)
- Then Click on Create Bucket

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, review that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ **Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐ **Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

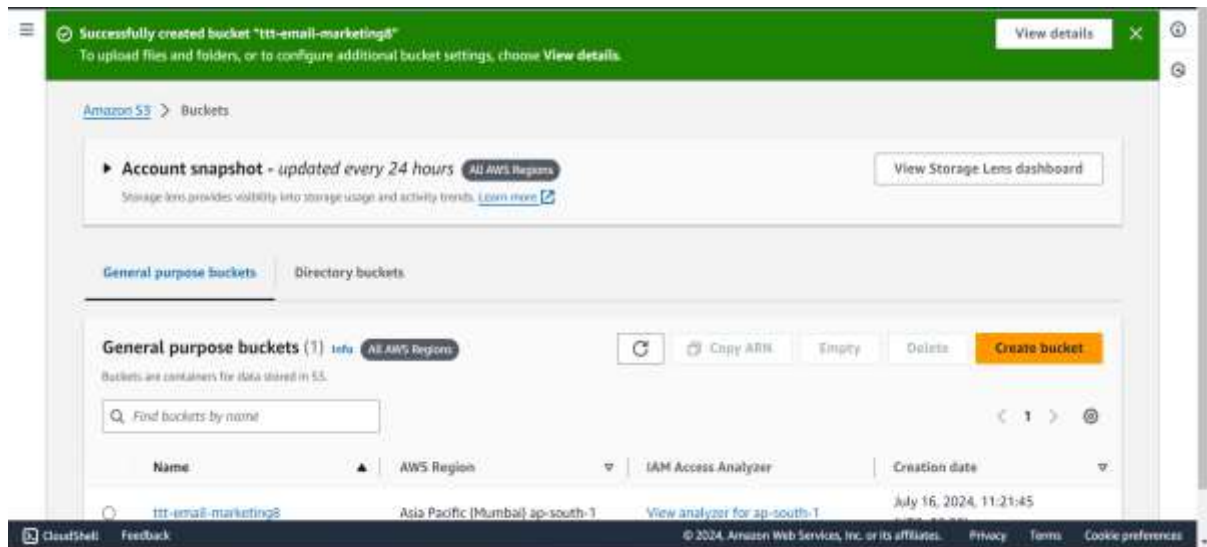
☐ **Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.

☐ **Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

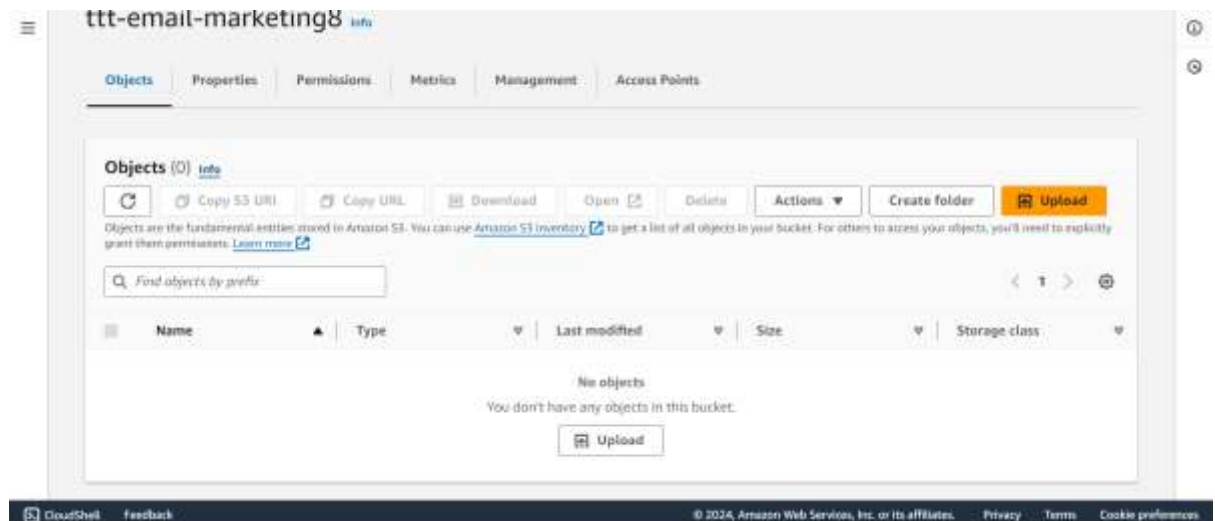
☐ **Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

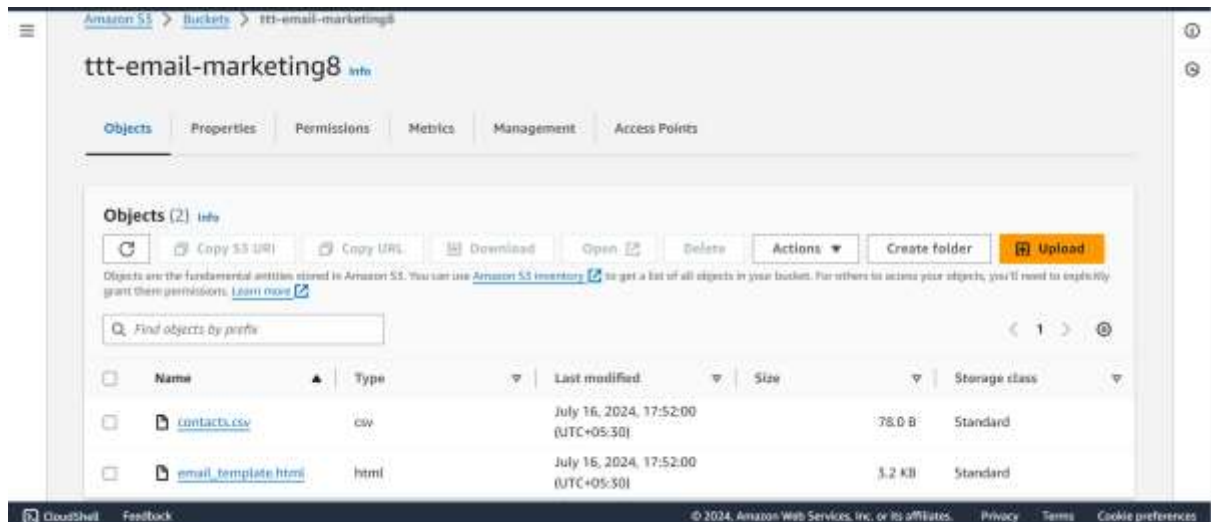
- You can view our newly created bucket in Amazon S3 under the “Buckets” section



- Click on the newly created bucket.
- Click the “Upload” button to add your website files.



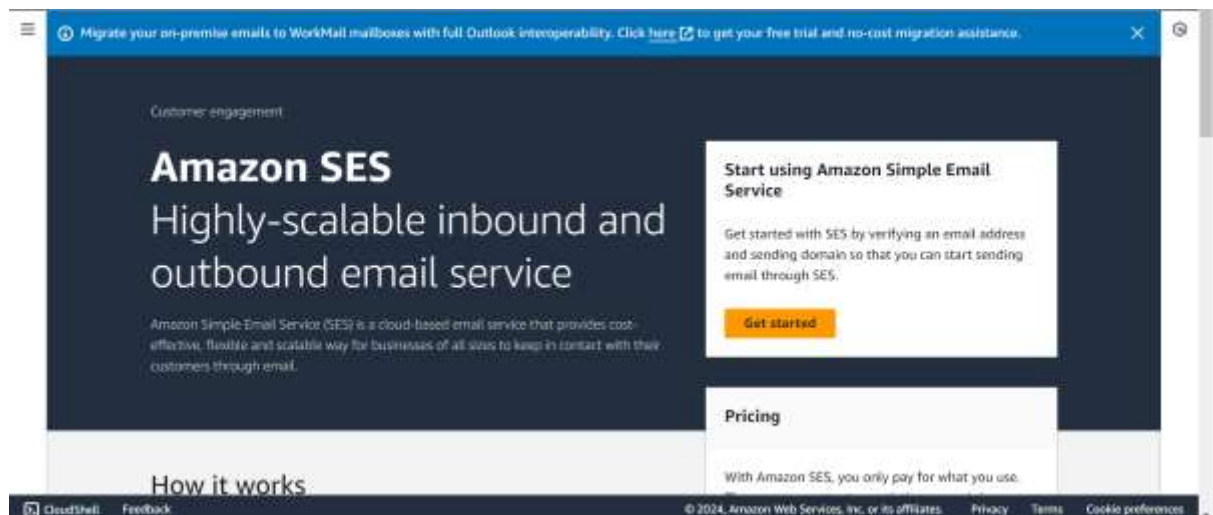
- You can upload all the files and folders of the website here.



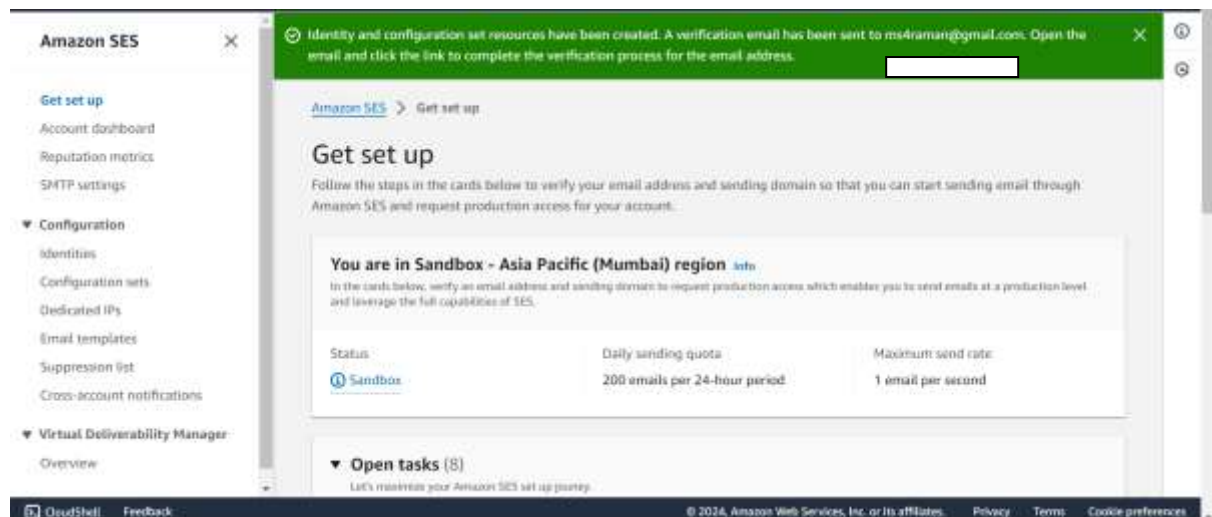
- And then click on Upload

Step 2: Next create a Simple Email Service for sending Email

- Navigate to SES in the AWS Console.
- Click the “Get Started” button



- And now we need to provide the email ID and domain name, and then create it



- Then you will get a Email for verification and you need to click that link then it will redirect to an message

Congratulations!

You have successfully verified an email address. You can now start sending email from this address.

For new Amazon SES users — If you have not yet applied for a sending limit increase, then you are still in the [sandbox environment](#), and you can only send email to addresses that have been verified. To verify a new email address or domain, see the **Identity Management** section of the [Amazon SES console](#).

For new Amazon Pinpoint users — If you have not yet applied for a sending limit increase, then you are still in the [sandbox environment](#), and you can only send email to addresses that have been verified. To verify a new email address or domain, see the **Settings > Channels** page on the [Amazon Pinpoint console](#).

If you have already been approved for a sending limit increase, then you can start sending email to non-verified addresses.

Thank you for using Amazon Web Services!

[Go to the Amazon SES detail page.](#)

Introducing Amazon Q.
Receive guidance, get troubleshooting tips,
and learn about AWS services and
capabilities.

AWS Business Support

What's New with AWS

- Now in completed task you can see the whether it is verified or not.

Amazon SES

Get set up

Account dashboard

Reputation metrics

SMTP settings

▼ Configuration

Identities

Configuration sets

Dedicated IPs

Email templates

Suppression list

Cross-account notifications

▼ Virtual Deliverability Manager

Overview


request permission access

123

View all tasks

▼ Completed tasks (1)

Tasks that have been successfully completed



Verified

Verify email address

ms4raman@gmail.com

Check your inbox to verify address.

View all identities

CloudShell

Feedback

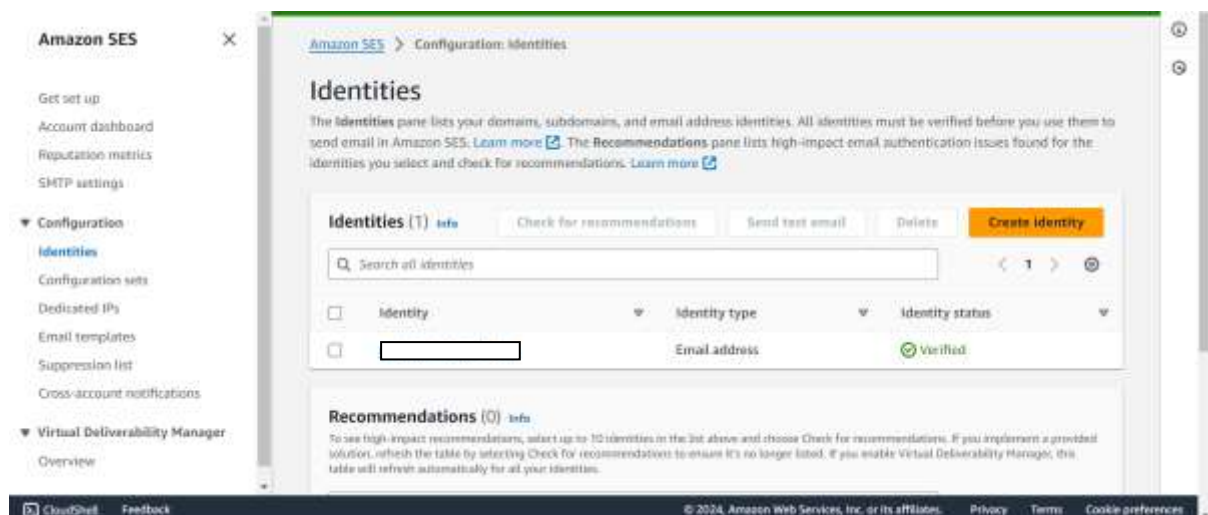
© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

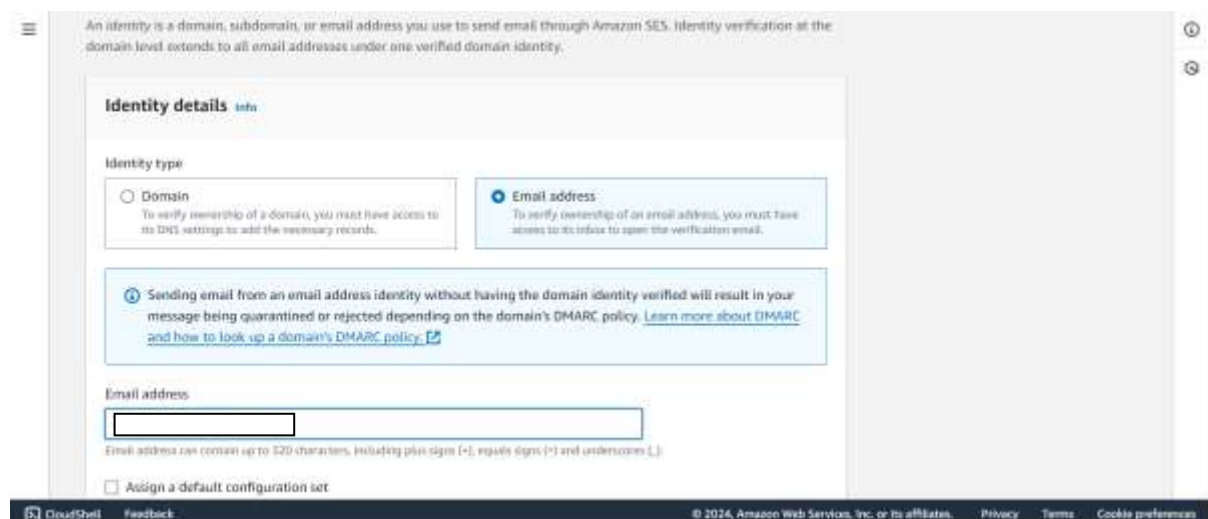
Terms

Cookie preferences

- Now that we have created the email ID, we need to send an email. To do this, we provided a contact. CSV file in S3 bucket that contains the email IDs of the recipients.
- So we are taking those email ID and create
- For that Go to create identity

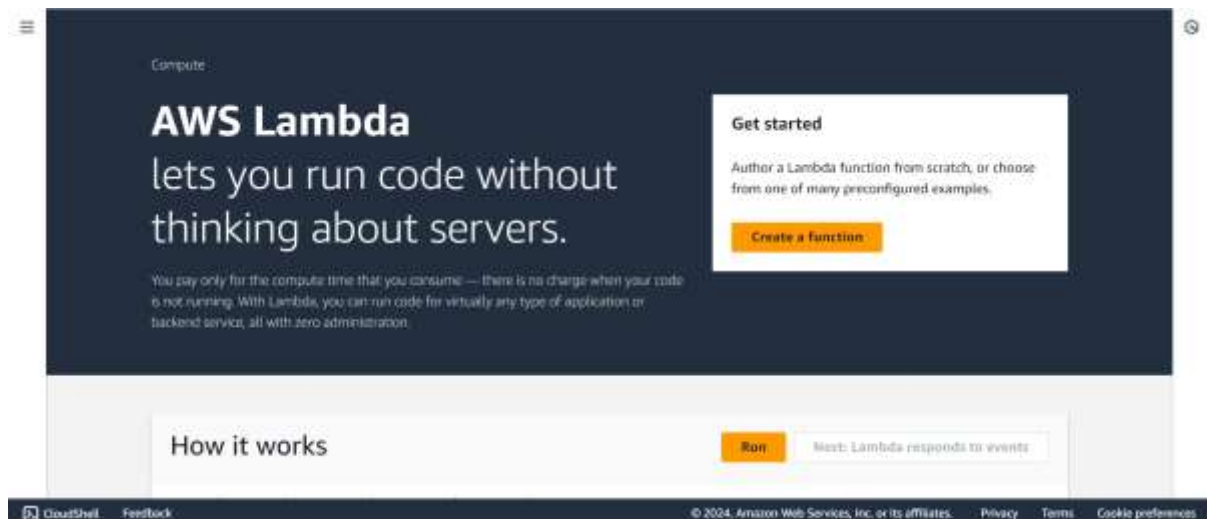


- You can have any number of identities/recipients. But in this instance, I only added one.
- And last click create identity.

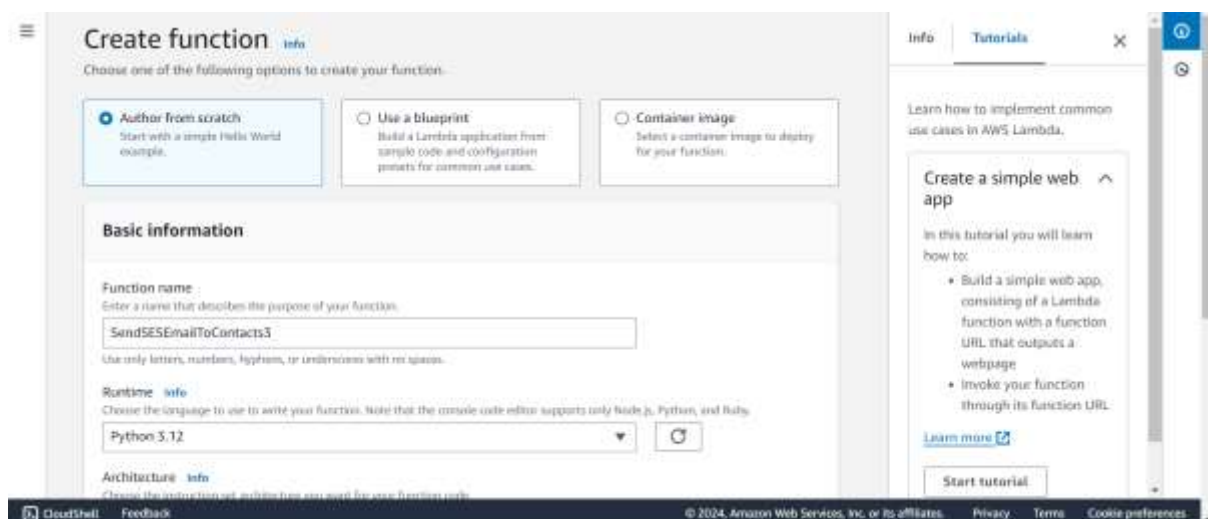


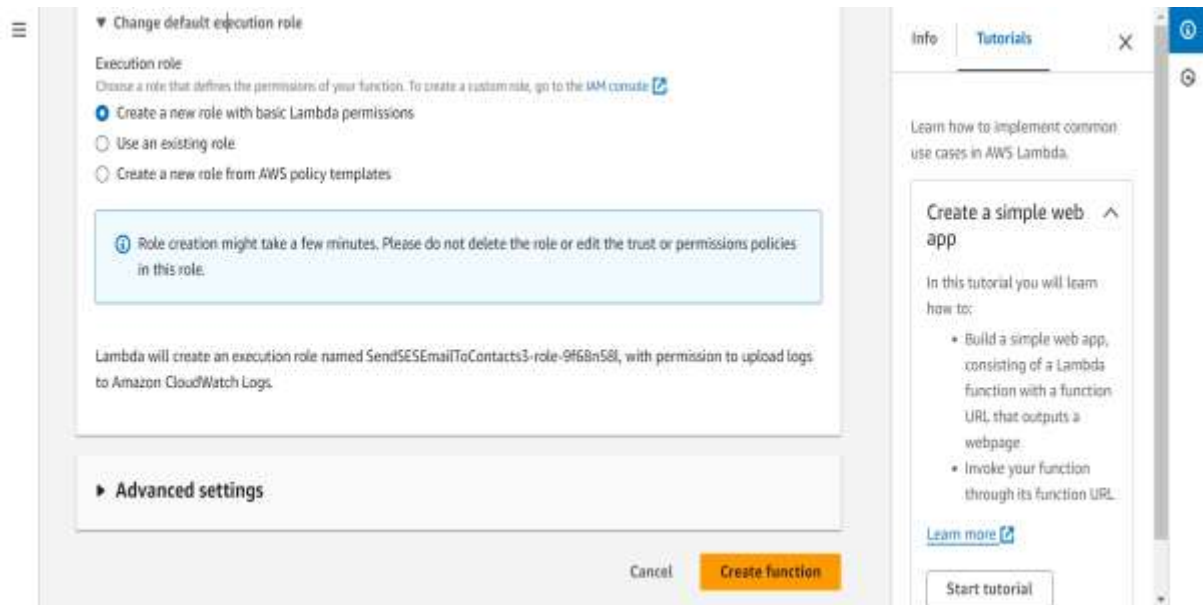
Step 3: Create Lambda Functions

- Navigate to AWS Lambda in the AWS Console.
- Click the “Create a Function” button

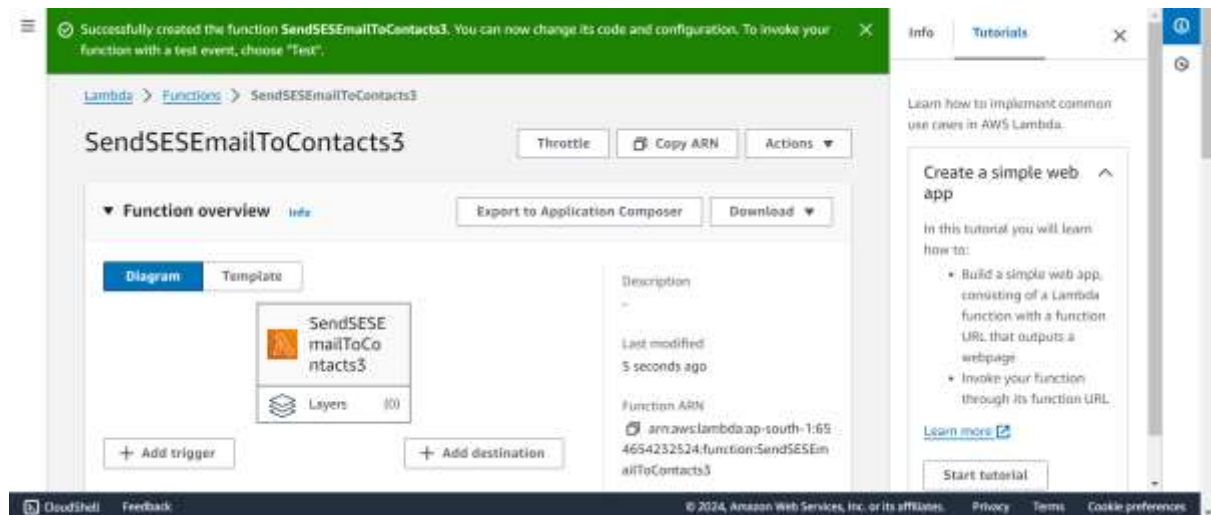


- Now we need to provide a name for the Lambda function, specify the language it should run.
- And also we can choose role also, then click on Create function.

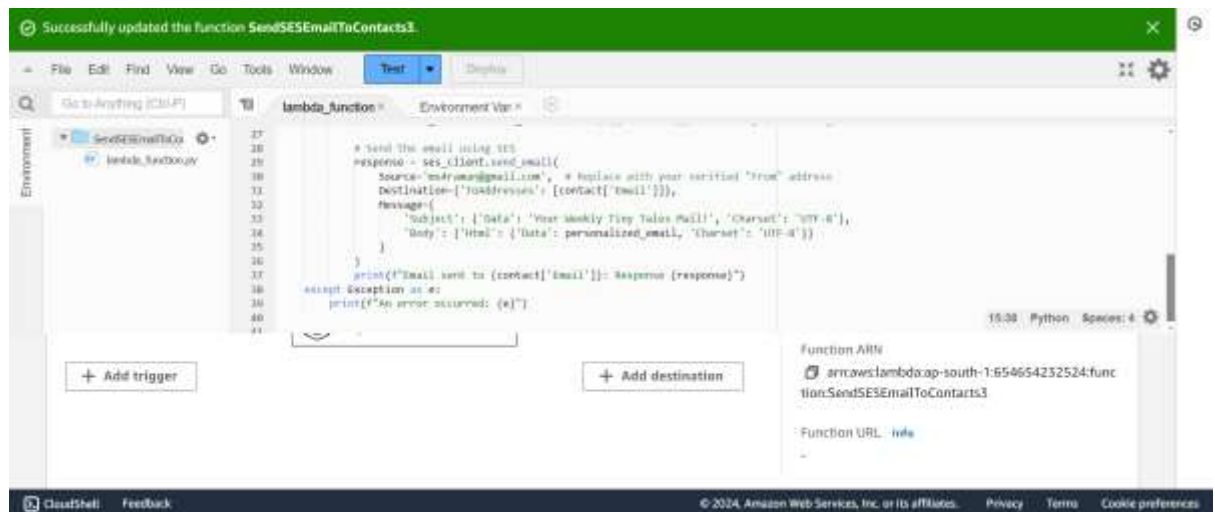
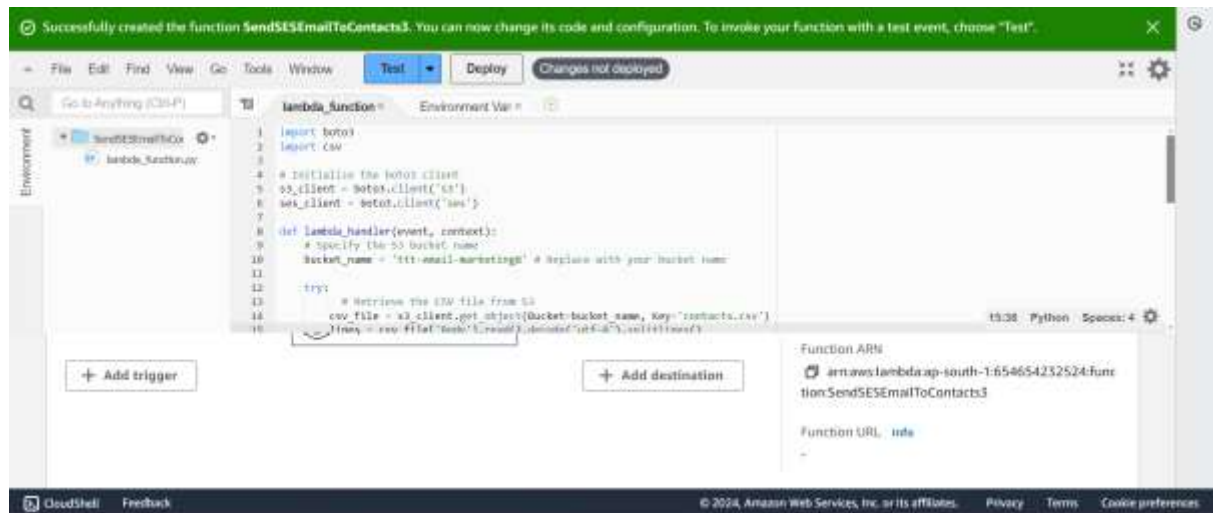




- Now we can see it get created



- So after creating the Lambda fuction we can change the code and configuration for our own.
- In this I have give one python code for the lambda Fuction



- After this we are checking whether this is working or not, for that we click the test
- Then need to give a Event name as well as one Event Json.
- And then click on Save

Configure test event

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

☒ Create new event ☐ Edit saved event

Event name

TestSendEmail

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

☒ Private
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

☐ Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Cancel Invoke Save

Event sharing settings

☒ Private
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

☐ Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

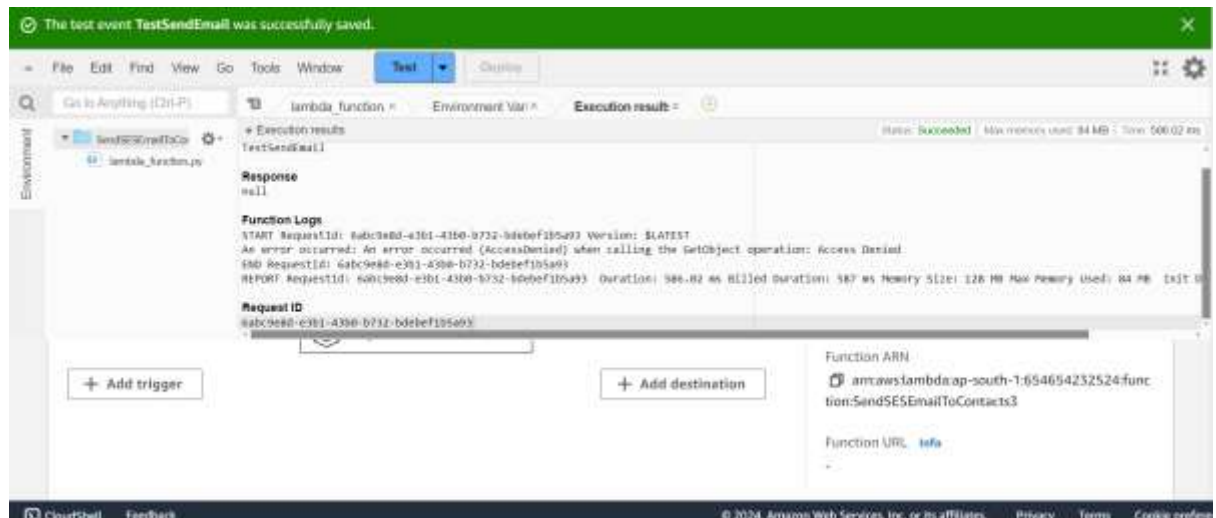
hello-world

Event JSON [Format JSON](#)

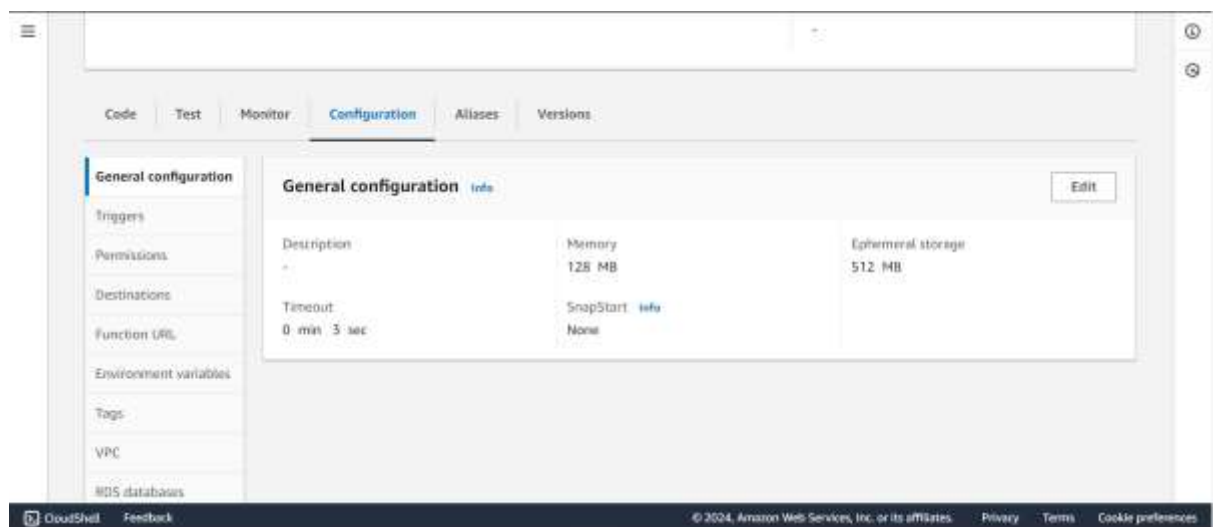
```
1 {  
2   "comment": "Generic test event for scheduled Lambda execution. The function does not use this event data.",  
3   "test": true  
4 }  
5
```

Cancel Invoke Save

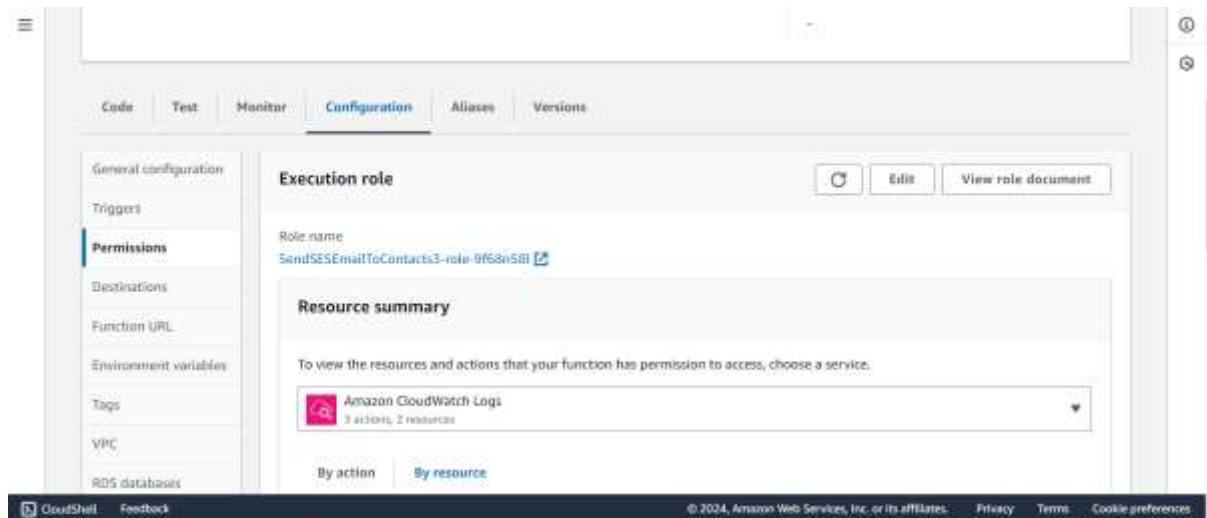
- After save, you need to click Deploy for whether our this Test event working or not.



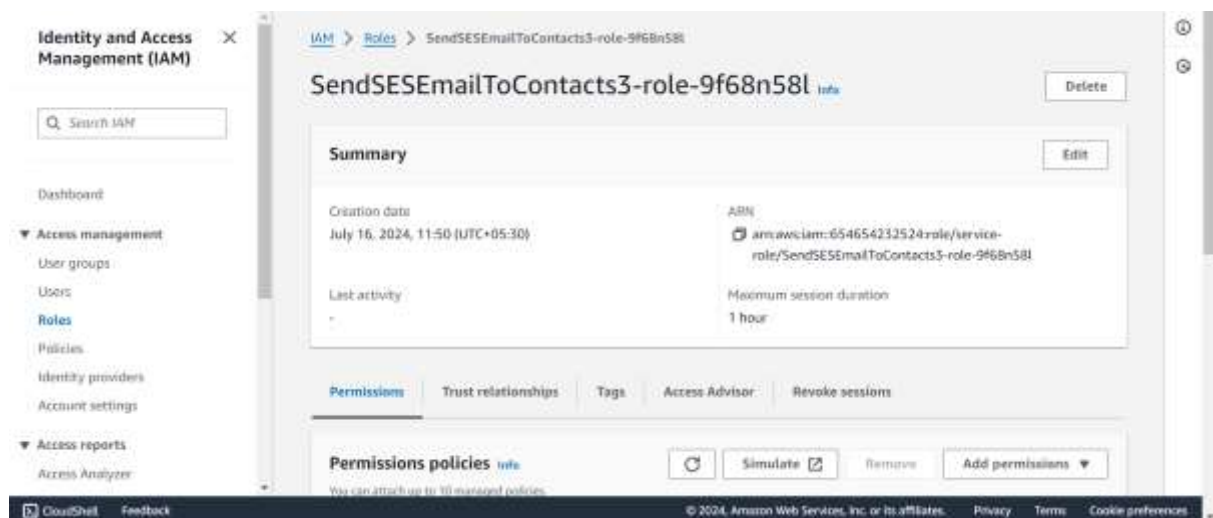
- But after Deploy we can see that it will show “Access Denied”.
- Because the Lambda Function is not having permission to do this, so because of that we need to make policies for that.
- Now click on Function, and go to Configuration



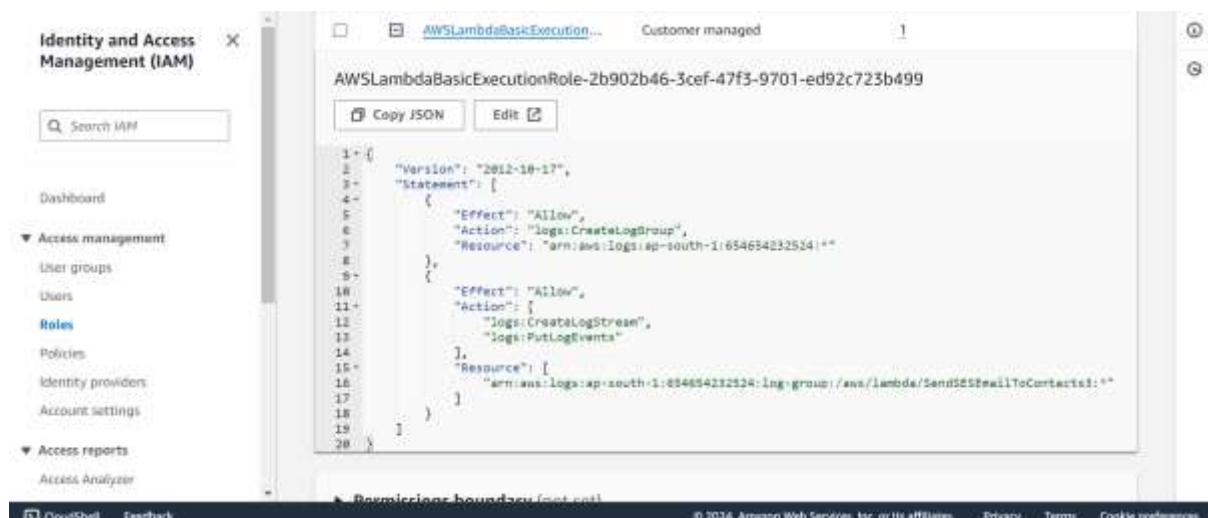
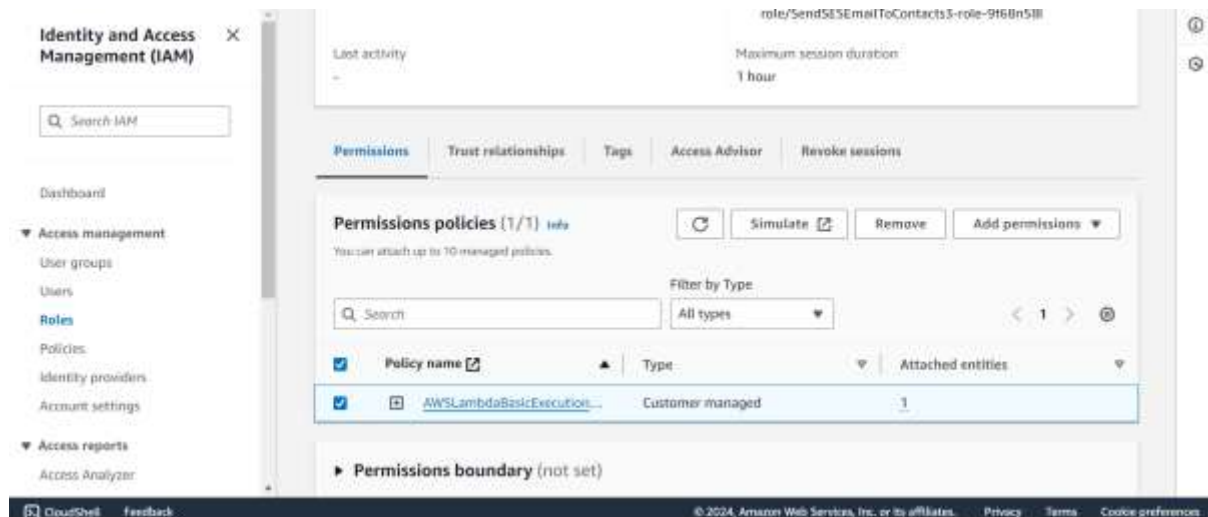
- Then Go to the Permissions
- Click on that Role name or that SendSESEmailTOContact3-role



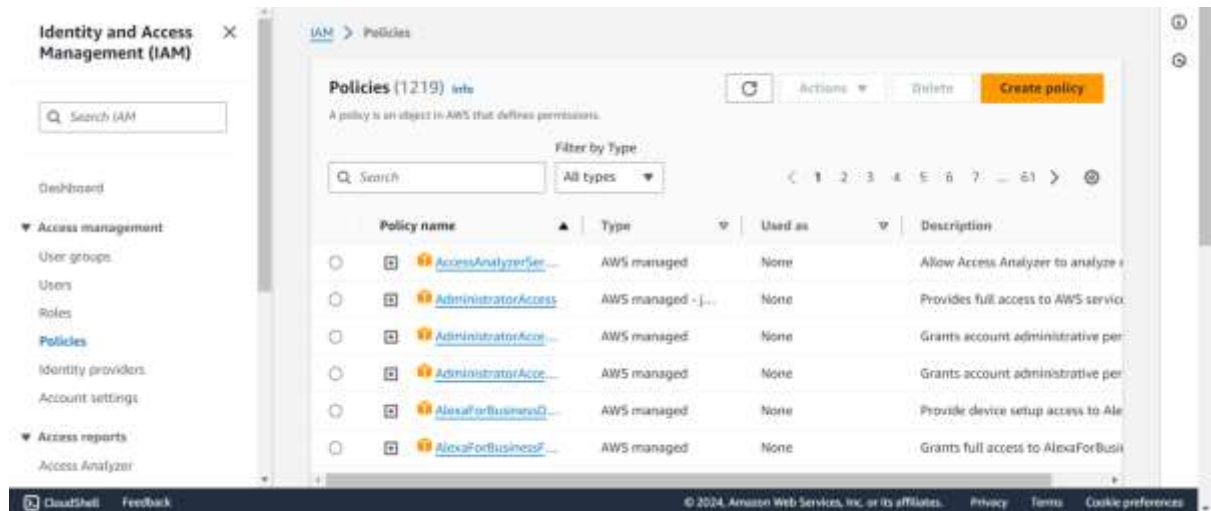
- Now Go to Roles



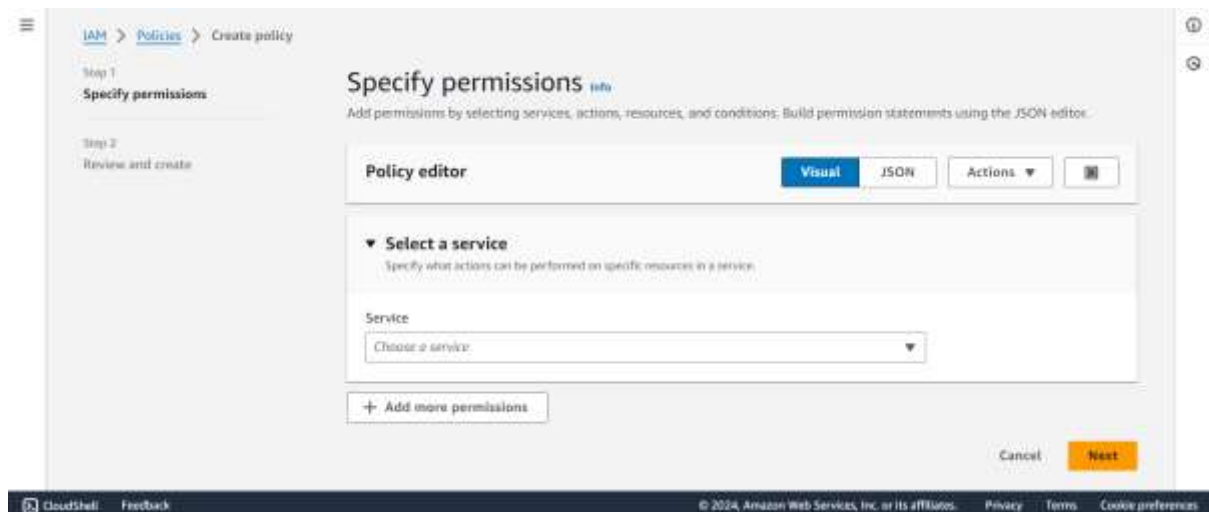
- Now, in roles, we need to check which role is assigned to this SES. It only has CloudWatch support.
- Because of that, we need to create a new policy.



- Now Go to policies and make a new policy for this
- Click on Create Policy



- After that you need to specify which Policy Editor need to choose
- Click on Json



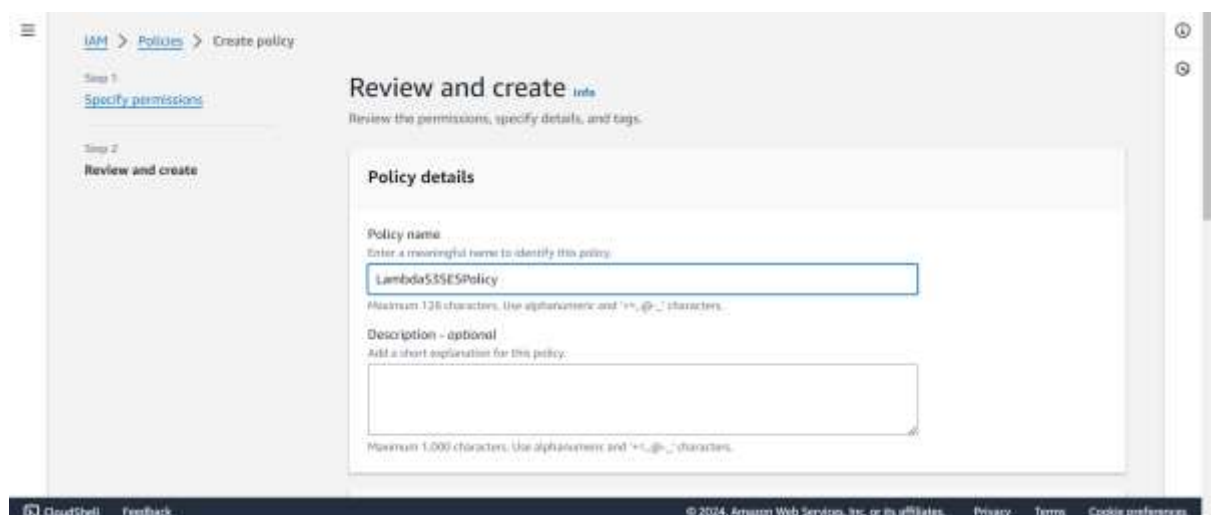
- And now need to make a code and put that into the JSON



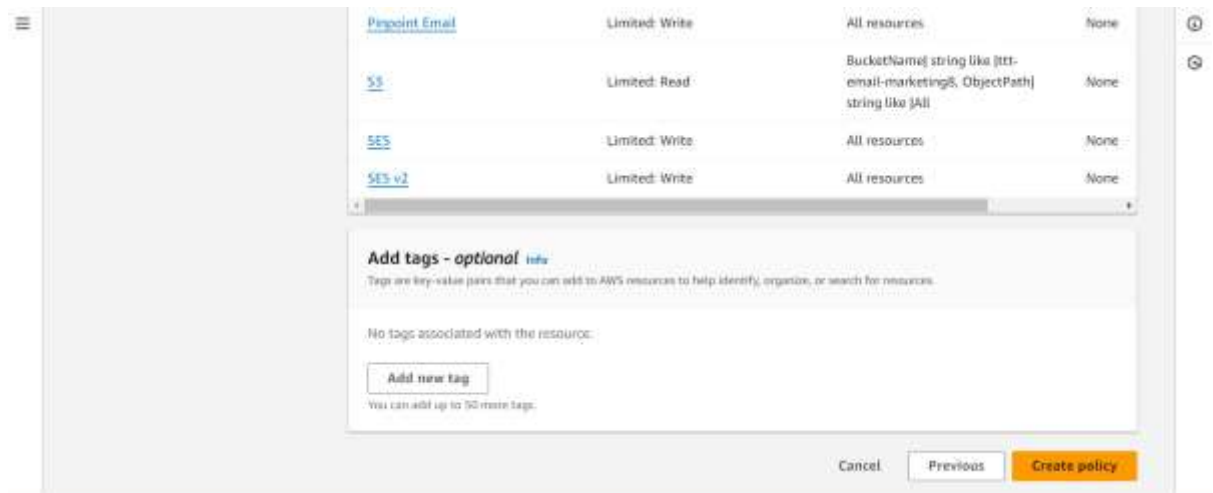
- Click on Next



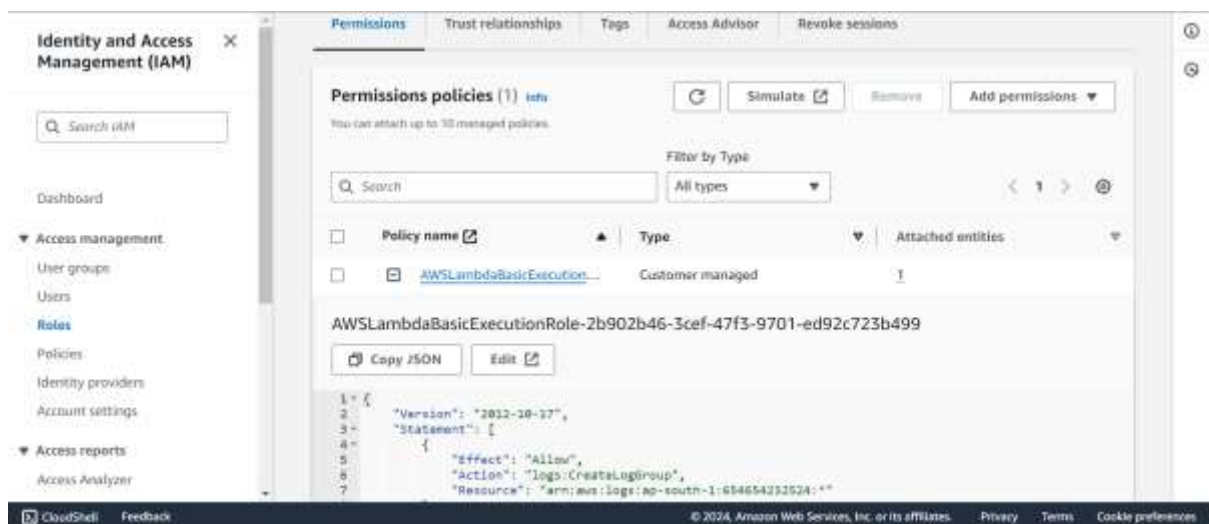
- In Review and Create we need to give a name for that (LambdaS3SESPolicy)



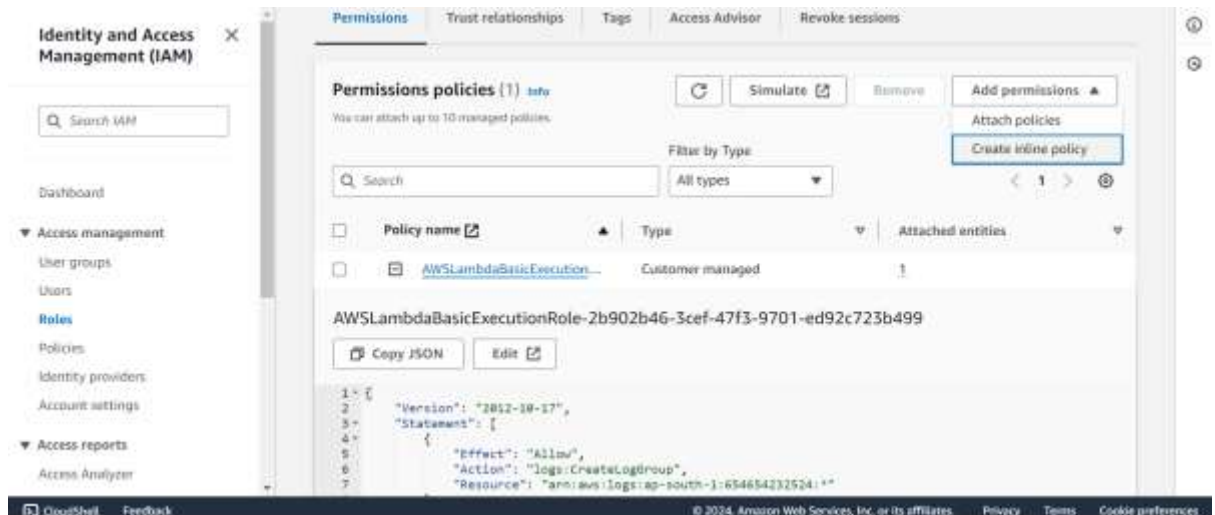
- And now click on Create Policy



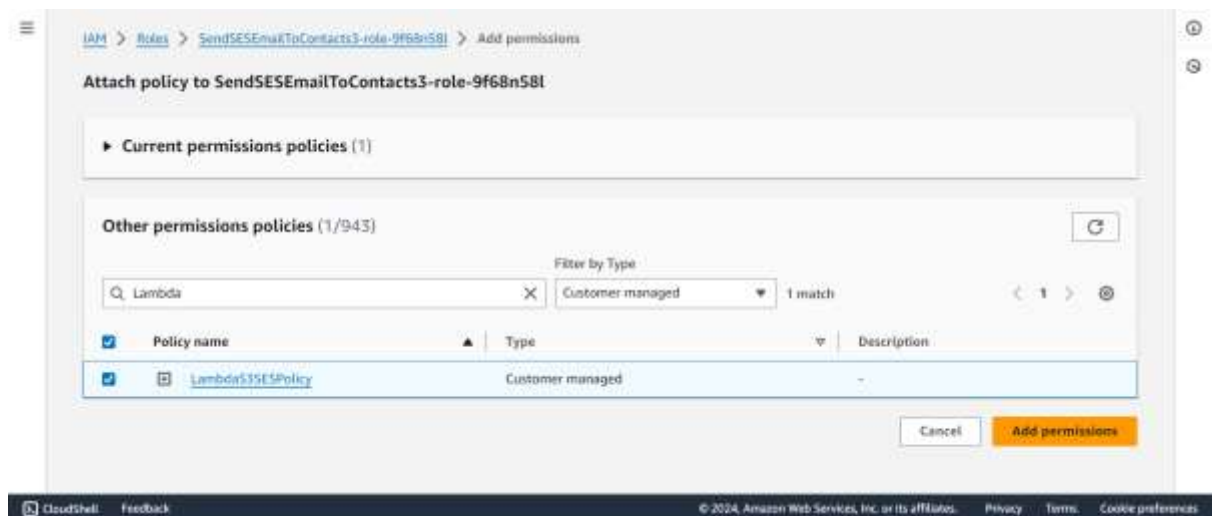
- Now need to add this permission into Policy
- For that Click on Add Permission



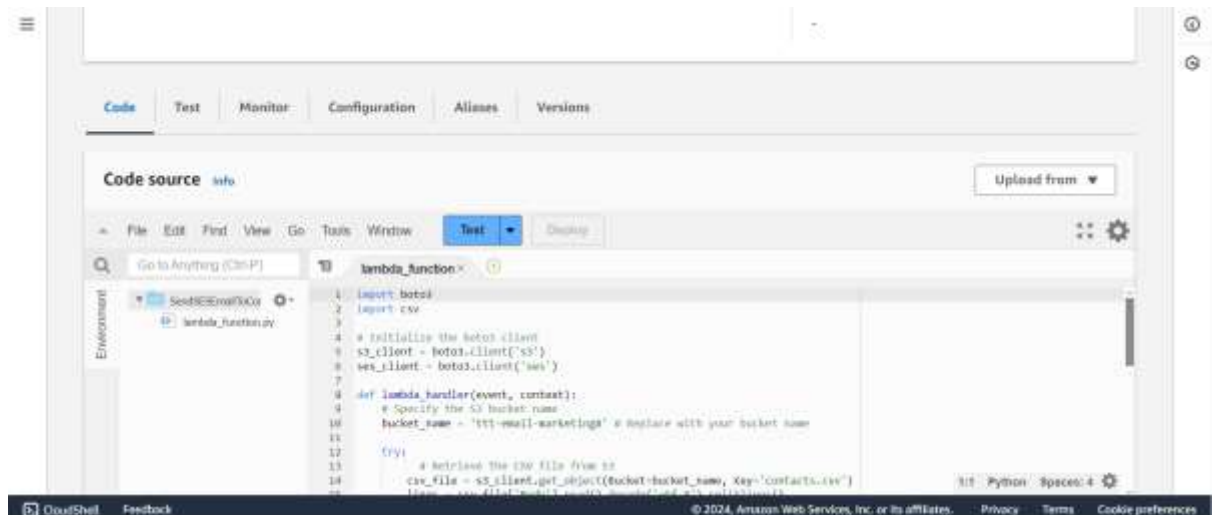
- And now click on Create Inline Policy



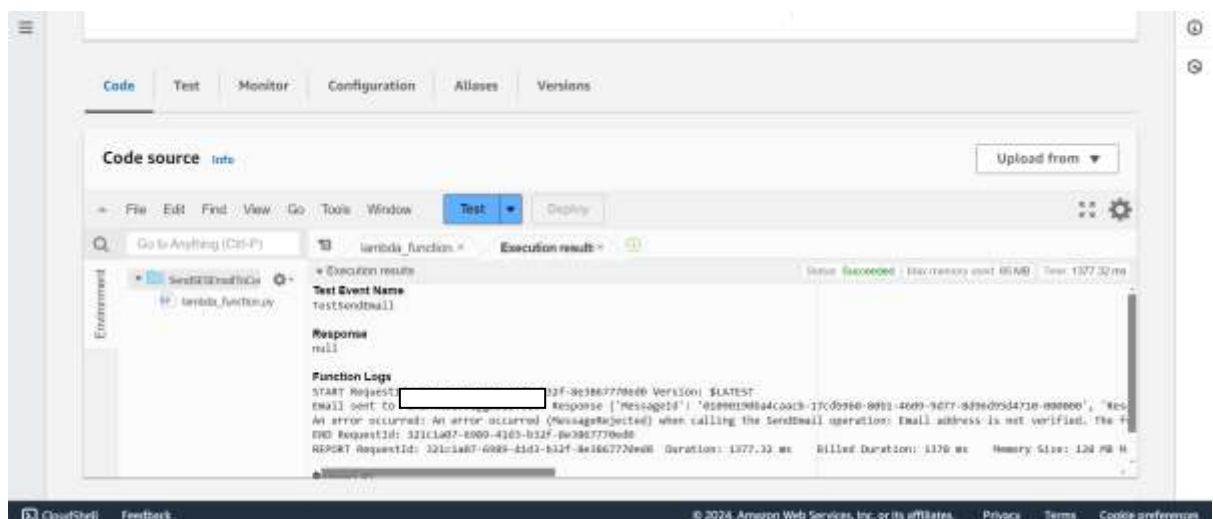
- Click that LamdaS3SESPolicy and Click on Add Permission



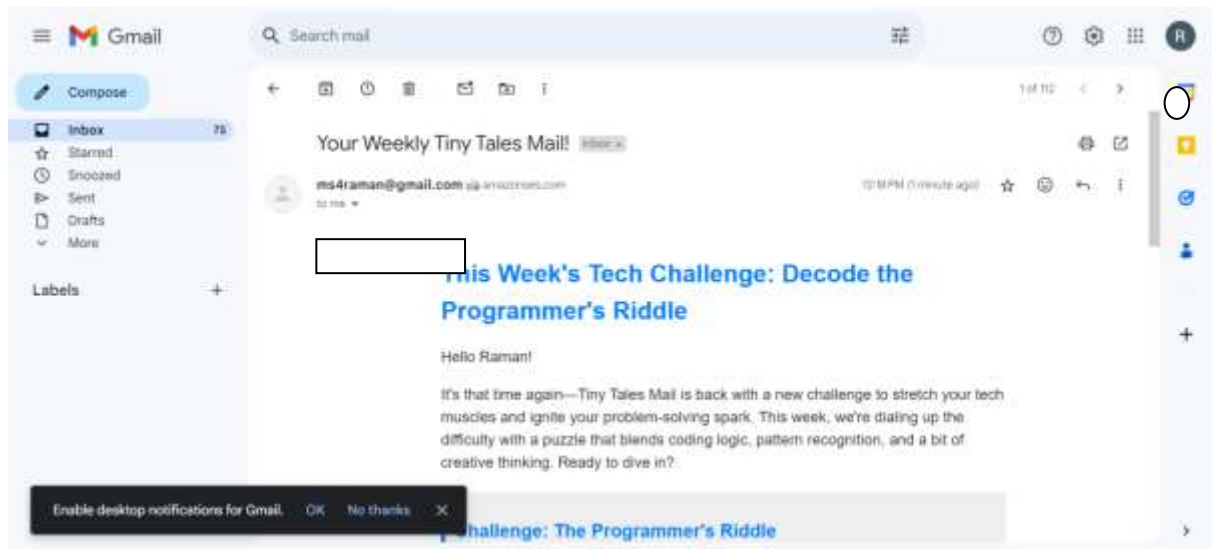
- After that again go the The Test and try once again



- Now you get a message like its success or you won't get the message called Access Denied



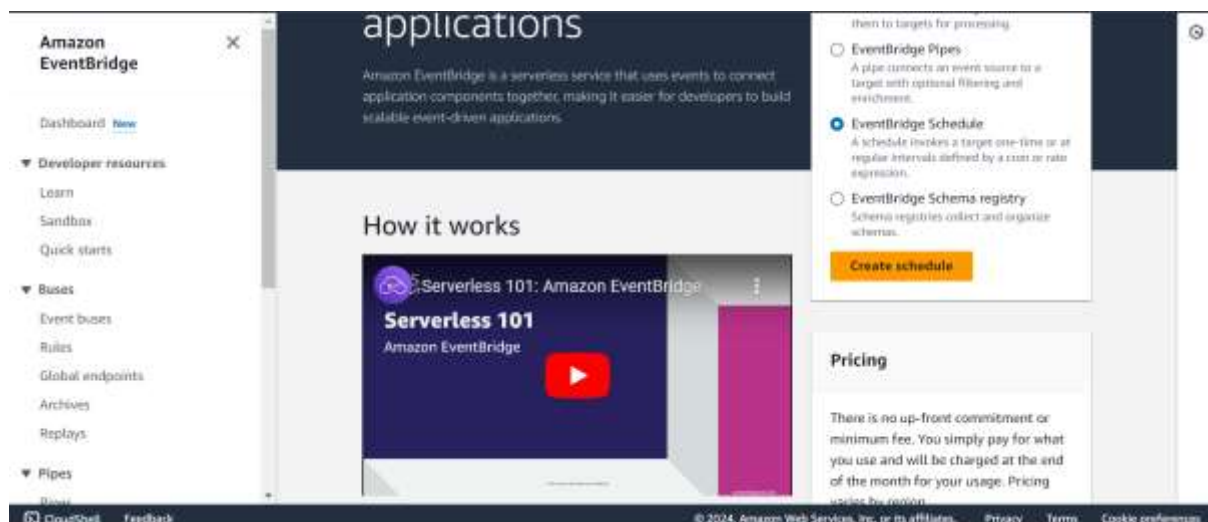
- Now, check your email to see if you received the email or not.



This will be the Email you will be Getting.

Step 4: Create Amazon EventBridge

- Navigate to Amazon EventBridge in the AWS Console.
- Click the EventBridge Schedule and Click the Create Schedule button



- Now we need to give a Schedule name

Step 1: Settings

Step 4: Review and create schedule

Schedule name

SendWeeklyEmail3

Use only letters, numbers, dashes, dots or underscores. Max 64 characters.

Description - optional

Enter description:

Maximum of 512 characters.

Schedule group

Each schedule needs to be placed in a schedule group. By default, a schedule is placed in the 'Default' group. You can also [create your own schedule group](#). You can only add tags to a schedule group, not a schedule.

default

Schedule pattern

Occurrence: info

You can define an one-time or recurring schedule.

One-time schedule

Recurring schedule

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- And next is the Schedule pattern, so here you can have once time schedule and also Recurring Schedule also.
- And specify the Year, Date and Time also the Time Zone
- Then Click on Next

Schedule pattern

Occurrence: info

You can define an one-time or recurring schedule.

One-time schedule

Recurring schedule

Date and time

The date and time to invoke the target.

2024/07/16

12:27

(UTC+05:30) Asia/Calcutta

YYYY/MM/DD

Use 24-hour format timestamp (HH:mm)

Time zone

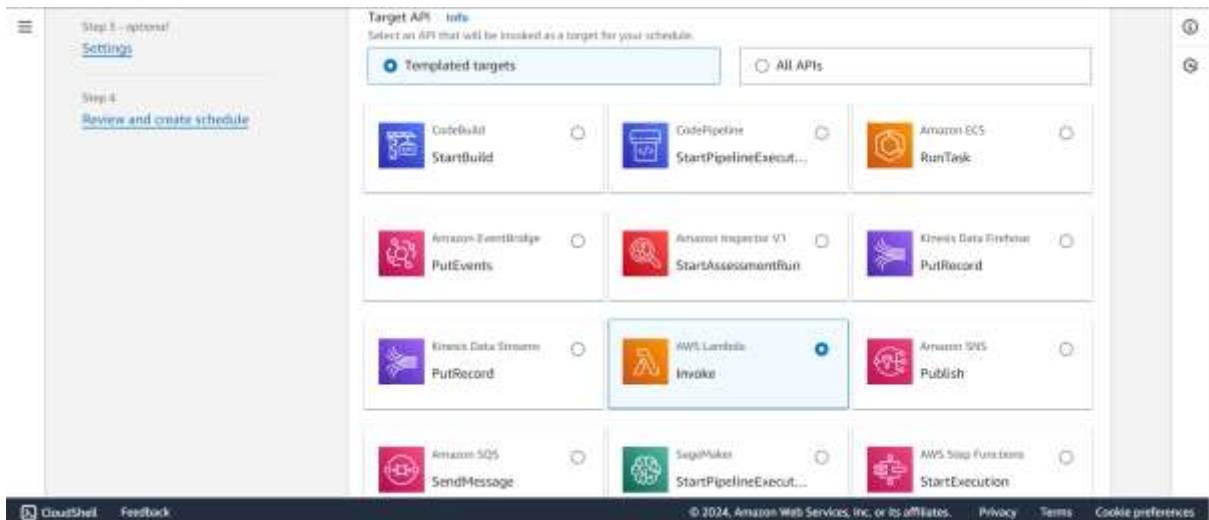
Flexible time window

If you choose a flexible time window, Schedule invokes your schedule within the time window you specify. For example, if you choose 15 minutes, your schedule runs within 15 minutes after the schedule start time.

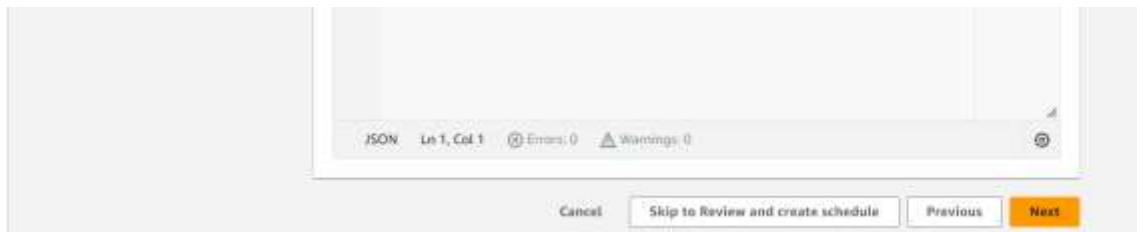
Select

Cancel Next

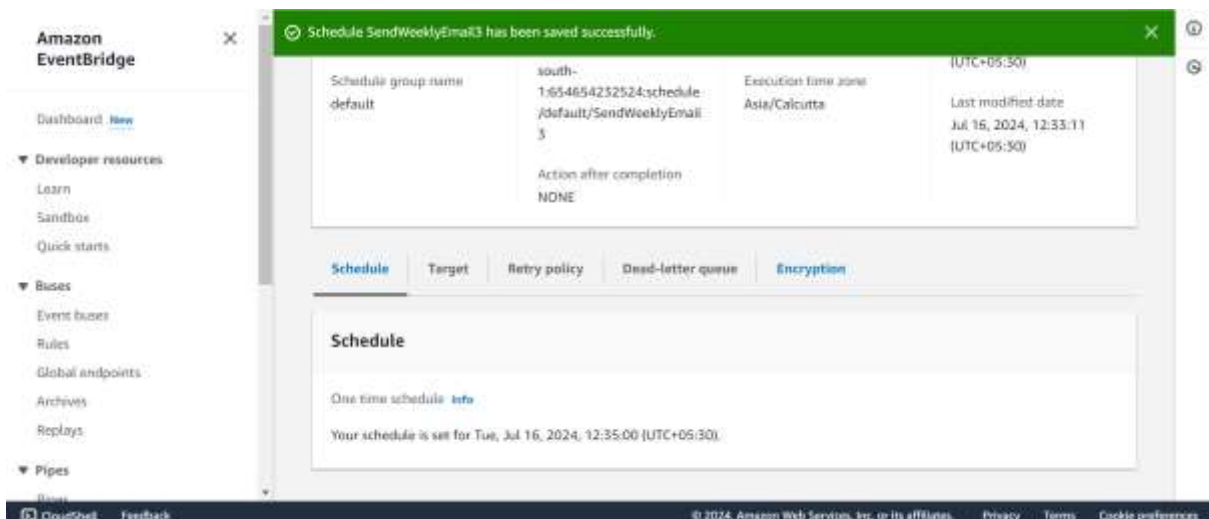
- After that we need to specify which Target Api want to choose , so here am choosing the AWS Lambda.



- Make Default in Settings and then atleast click Next and Create

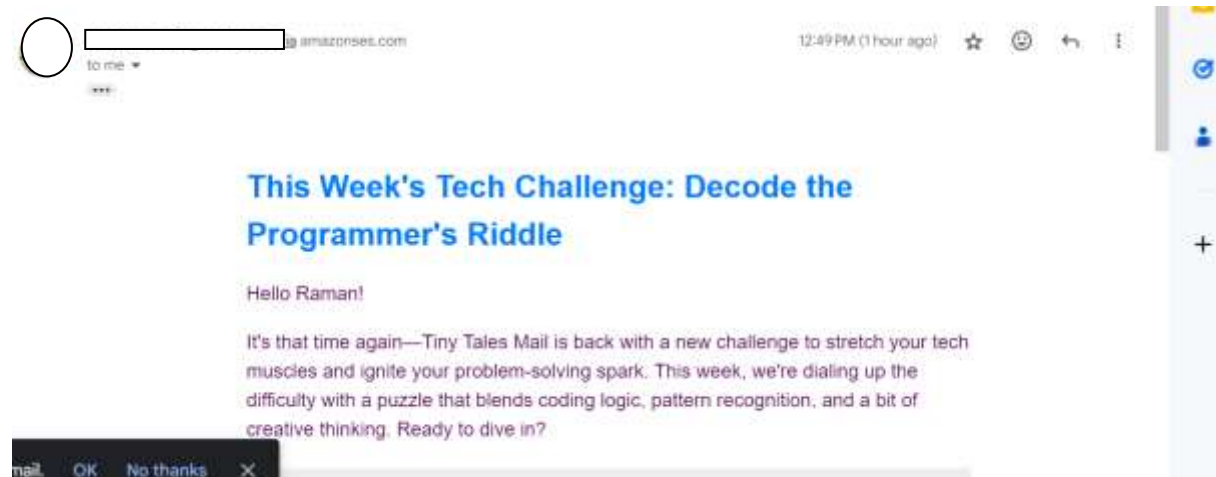


- So Successfully I have created the EventBridge and wait for the Result in Email



- Tadddaaaa! Horrayyyyyy we go the OutPut

-



Here timing might change here because this eventbridge step I tried two times and the second time I couldn't take snap